Course Title : Computer Architecture

Course code: CCE2209

Instructor : Dr. Mahmoud Alshewimy

General information

Course	:Computer Architecture
Instructor	: Dr. Mahmoud Alshewimy
Email	: malshewimy@gmail.com
Lecture time	: Saturday (10:30am – 12:15 am) & Thursday 08:30am – 10:15 am.
Office hours	:
Teaching Assistant:	
Email	:
Office hours	: Discuss with the TA

Course Objective

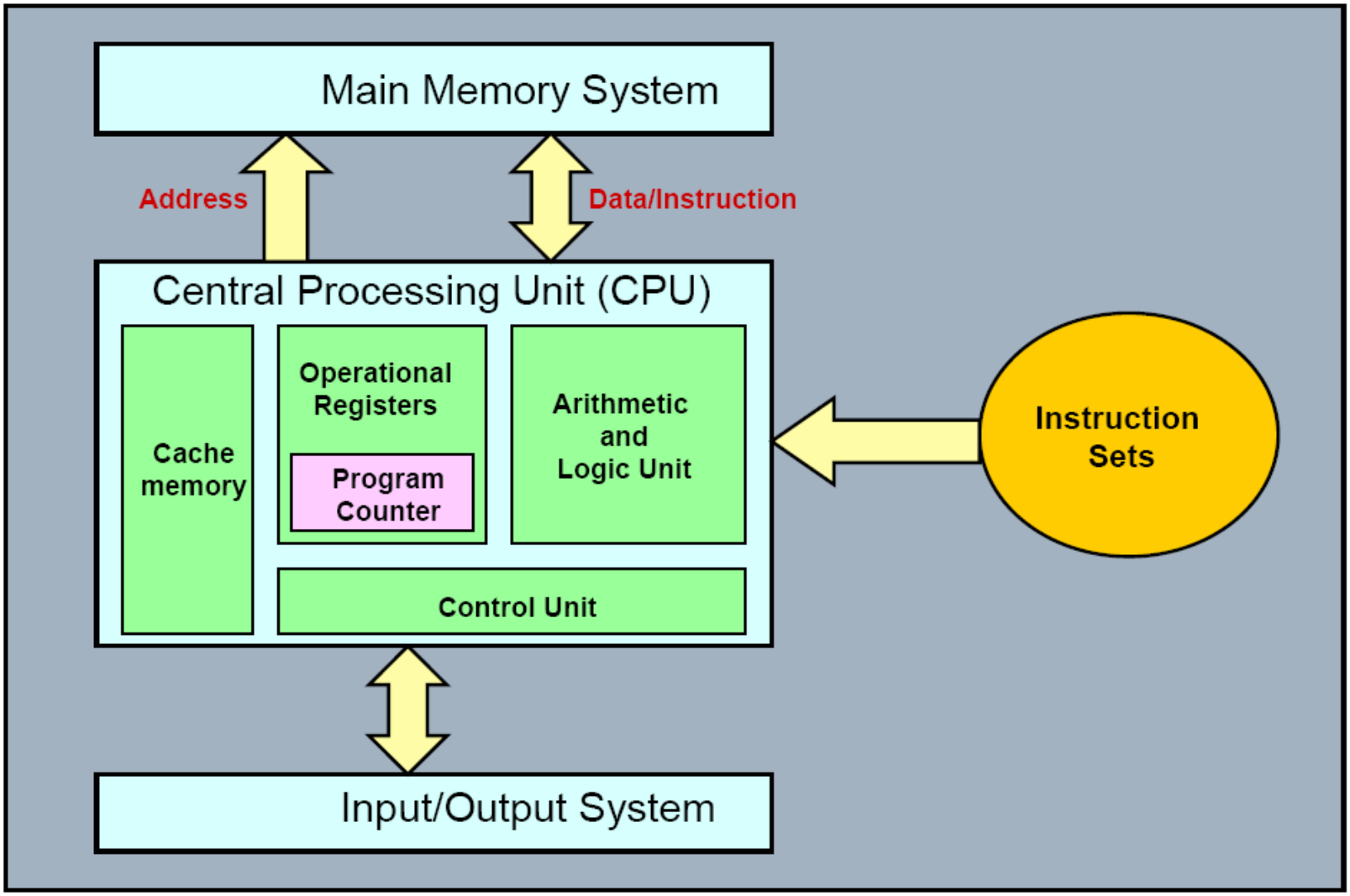
- Describe the general organization and architecture of computers.
- Identify computers' major components and study their functions.
- Introduce hardware design issues of modern computer architectures.
- Build the required skills to read and research the current literature in computer architecture.

Textbooks

- "Computer Organization," by Carl Hamacher, Zvonko Vranesic and Safwat Zaky. Fifth Edition McGraw-Hill, 2002.

- Lecture notes.

Content Coverage



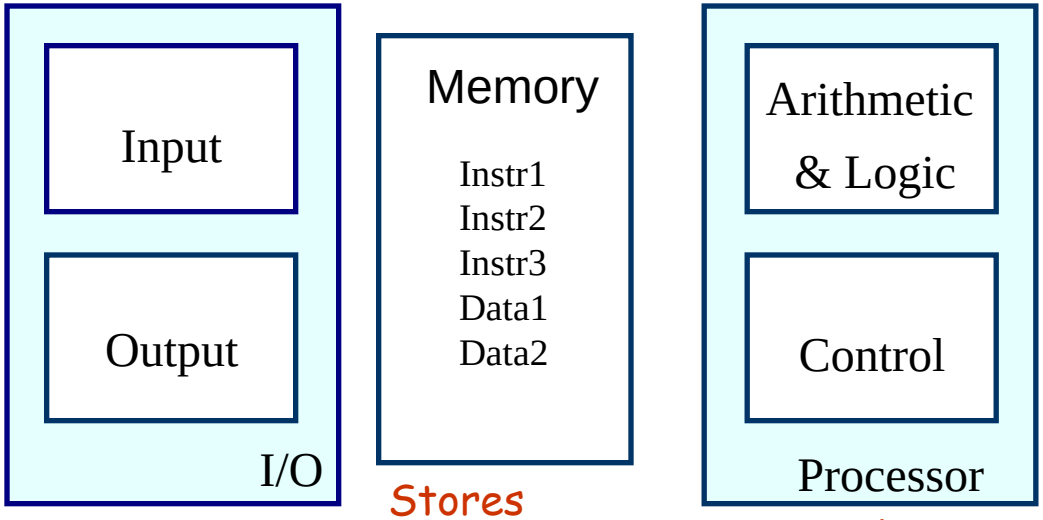
What is a computer?

- A computer is a sophisticated electronic calculating machine that:
  - Accepts input data,
  - Processes the data according to a list of internally stored instructions and
  - Produces the resulting output data.
- Functions performed by a computer are:
  - Accepting data to be processed as input.
  - Storing a list of instructions to process the data.
  - Processing the data according to the list of instructions.
  - Providing the results of the processing as output.
- What are the functional units of a computer?

Functional units of a computer

Input unit accepts information:
- Human operators,
- Electromechanical devices (keyboard)
- Other computers

Arithmetic and logic unit(ALU):
- Performs the desired operations on the input information as determined by instructions in the memory



Output unit sends results of processing:
- To a monitor display,
- To a printer

Stores information:
- Instructions,
- Data

Control unit coordinates various actions
- Input,
- Output
- Processing

Information in a computer -- *Instructions*

- Instructions specify commands to:
  - Transfer information within a computer (e.g., from memory to ALU)
  - Transfer of information between the computer and I/O devices (e.g., from keyboard to computer, or computer to printer)
  - Perform arithmetic and logic operations (e.g., Add two numbers, Perform a logical AND).
- A sequence of instructions to perform a task is called a program, which is stored in the memory.
- Processor fetches instructions that make up a program from the memory and performs the operations stated in those instructions.
- What do the instructions operate upon?

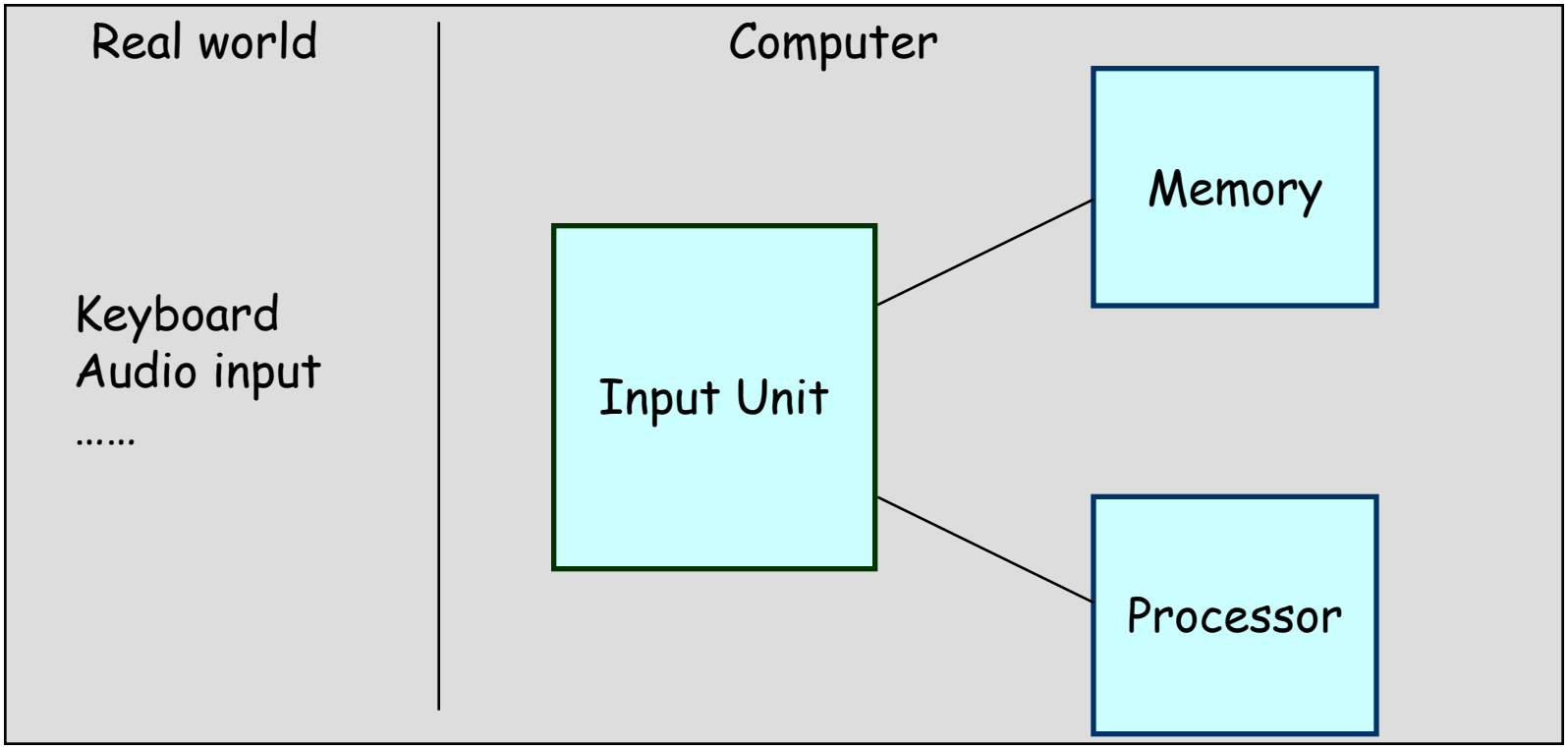
Information in a computer -- **Data**

- Data are the "operands" upon which instructions operate.
- Data could be:
  - Numbers,
  - Encoded characters.
- Data, in a broad sense means any digital information.
- Computers use data that is encoded as a string of binary digits called bits.

Input unit

Binary information must be presented to a computer in a specific format. This task is performed by the input unit:

- Interfaces with input devices.
- Accepts binary information from the input devices.
- Presents this binary information in a format expected by the computer.
- Transfers this information to the memory or processor.



Memory unit

- Memory unit stores instructions and data.
  - Recall, data is represented as a series of bits.
  - To store data, memory unit thus stores bits.
- Processor reads instructions and reads/writes data from/to the memory during the execution of a program.
  - In theory, instructions and data could be fetched one bit at a time.
  - In practice, a group of bits is fetched at a time.
  - Group of bits stored or retrieved at a time is termed as "word"
  - Number of bits in a word is termed as the "word length" of a computer.
- In order to read/write to and from memory, a processor should know where to look:
  - "Address" is associated with each word location.

Memory unit (contd..)

- Processor reads/writes to/from memory based on the memory address:
  - Access any word location in a short and fixed amount of time based on the address.
  - Random Access Memory (RAM) provides fixed access time independent of the location of the word.
  - Access time is known as "Memory Access Time".
- Memory and processor have to "communicate" with each other in order to read/write information.
  - In order to reduce "communication time", a small amount of RAM (known as Cache) is tightly coupled with the processor.
- Modern computers have three to four levels of RAM units with different speeds and sizes:
  - Fastest, smallest known as Cache
  - Slowest, largest known as Main memory.

Memory unit (contd..)

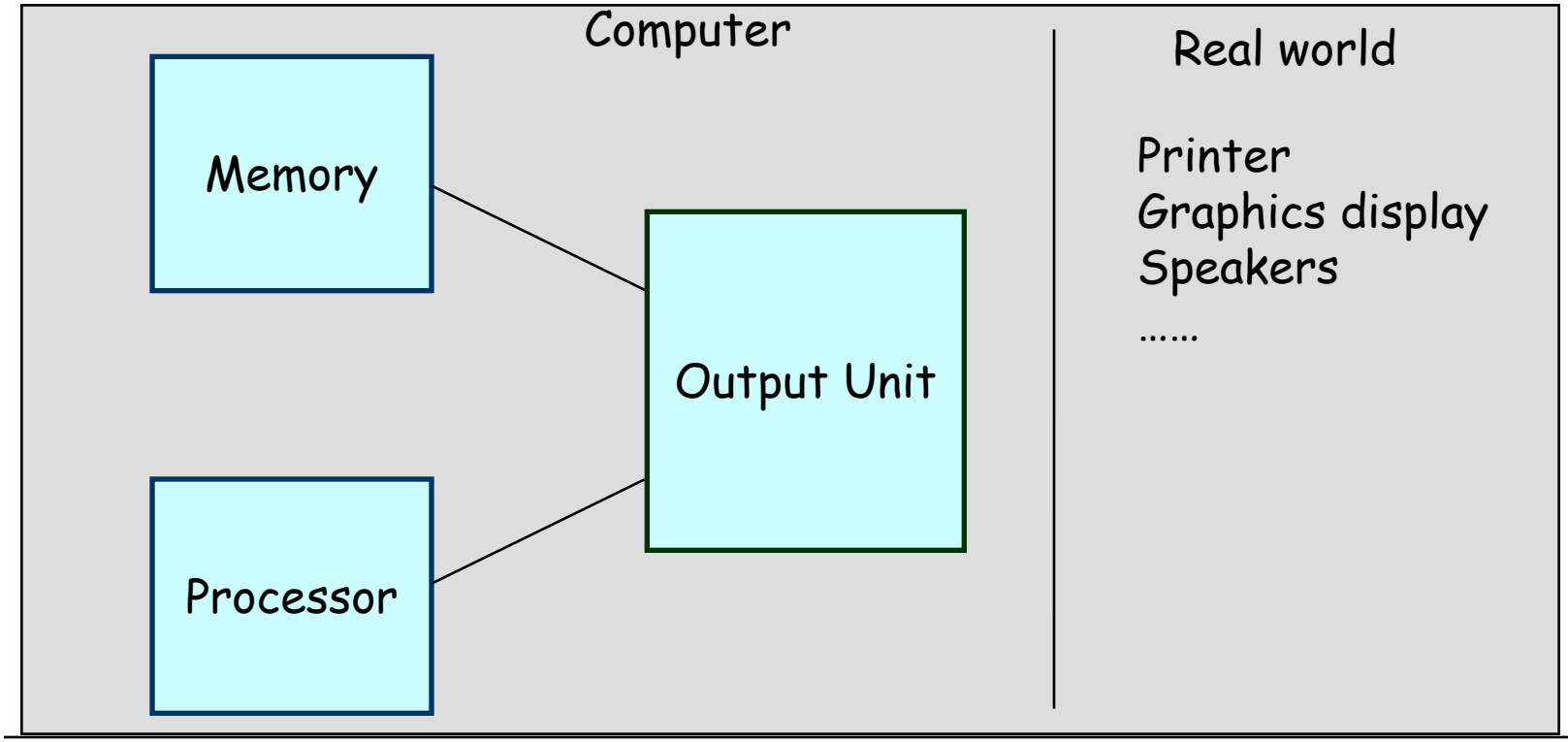
- Primary storage of the computer consists of RAM units.
  - Fastest, smallest unit is Cache.
  - Slowest, largest unit is Main Memory.
- Primary storage is insufficient to store large amounts of data and programs.
  - Primary storage can be added, but it is expensive.
- Store large amounts of data on secondary storage devices:
  - Magnetic disks and tapes,
  - Optical disks (CD-ROMS).
  - Access to the data stored in secondary storage in slower, but take advantage of the fact that some information may be accessed infrequently.
- Cost of a memory unit depends on its access time, lesser access time implies higher cost.

Arithmetic and logic unit (ALU)

- Operations are executed in the Arithmetic and Logic Unit (ALU).
  - Arithmetic operations such as addition, subtraction.
  - Logic operations such as comparison of numbers.
- In order to execute an instruction, operands need to be brought into the ALU from the memory.
  - Operands are stored in general purpose registers available in the ALU.
  - Access times of general purpose registers are faster than the cache.
- Results of the operations are stored back in the memory or retained in the processor for immediate use.

Output unit

- Computers represent information in a specific binary form. Output units:
  - Interface with output devices.
  - Accept processed results provided by the computer in specific binary form.
  - Convert the information in binary form to a form understood by an output device.

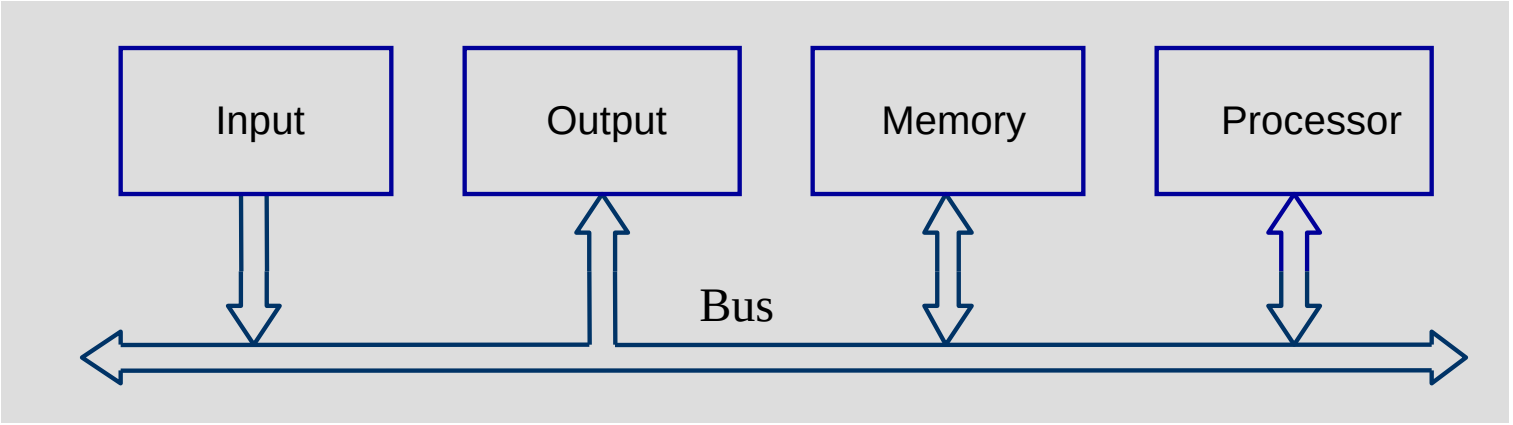


Control unit

- Operation of a computer can be summarized as:
  - Accepts information from the input units (Input unit).
  - Stores the information (Memory).
  - Processes the information (ALU).
  - Provides processed results through the output units (Output unit).
- Operations of Input unit, Memory, ALU and Output unit are coordinated by Control unit.
- Instructions control "what" operations take place (e.g. data transfer, processing).
- Control unit generates timing signals which determines "when" a particular operation takes place.
- Control unit can be represented as a state machine.

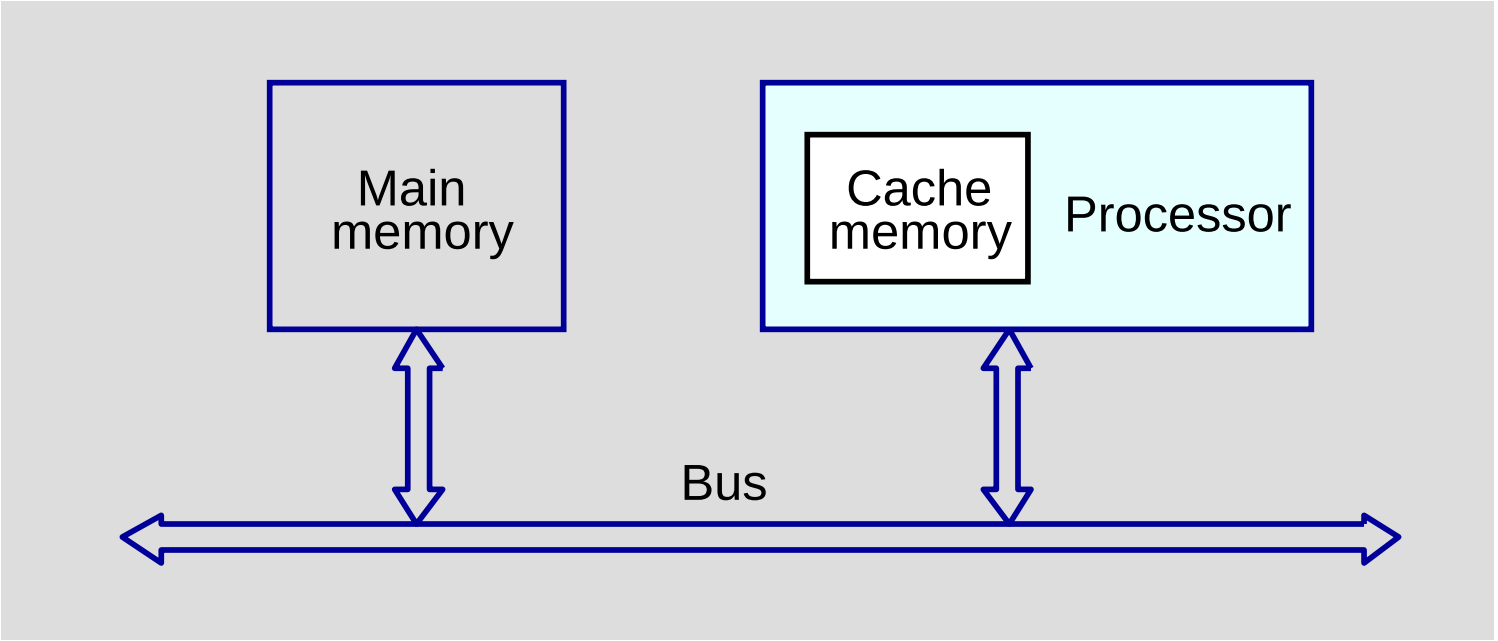
How are the functional units connected?

- For a computer to achieve its operation, the functional units need to communicate with each other.
- In order to communicate, they need to be connected.



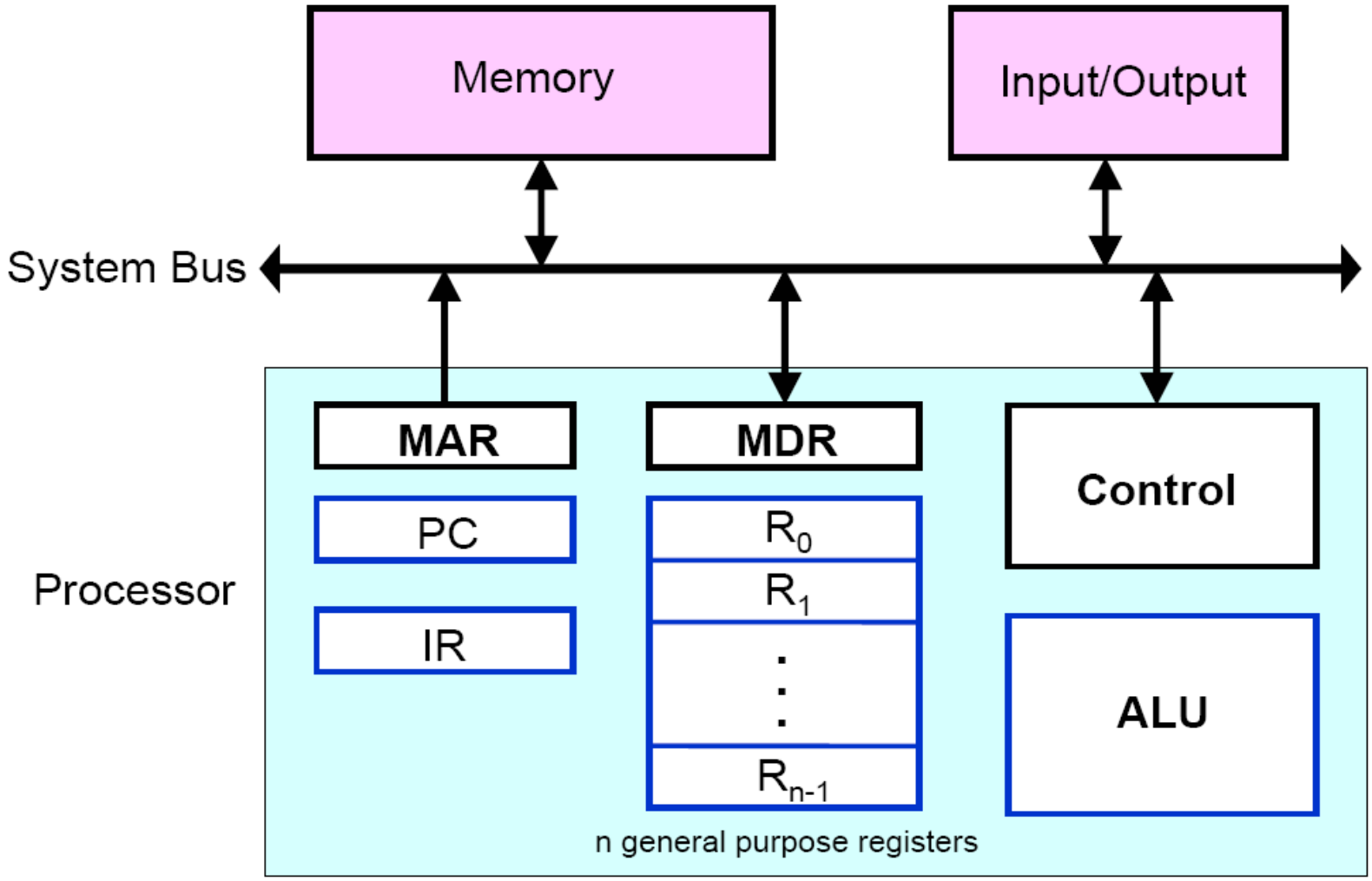
- Functional units may be connected by a group of parallel wires.
- The group of parallel wires is called a ***bus***.
- Each wire in a bus can transfer one bit of information.
- The number of parallel wires in a bus is equal to the word length of a computer

Organization of cache and main memory

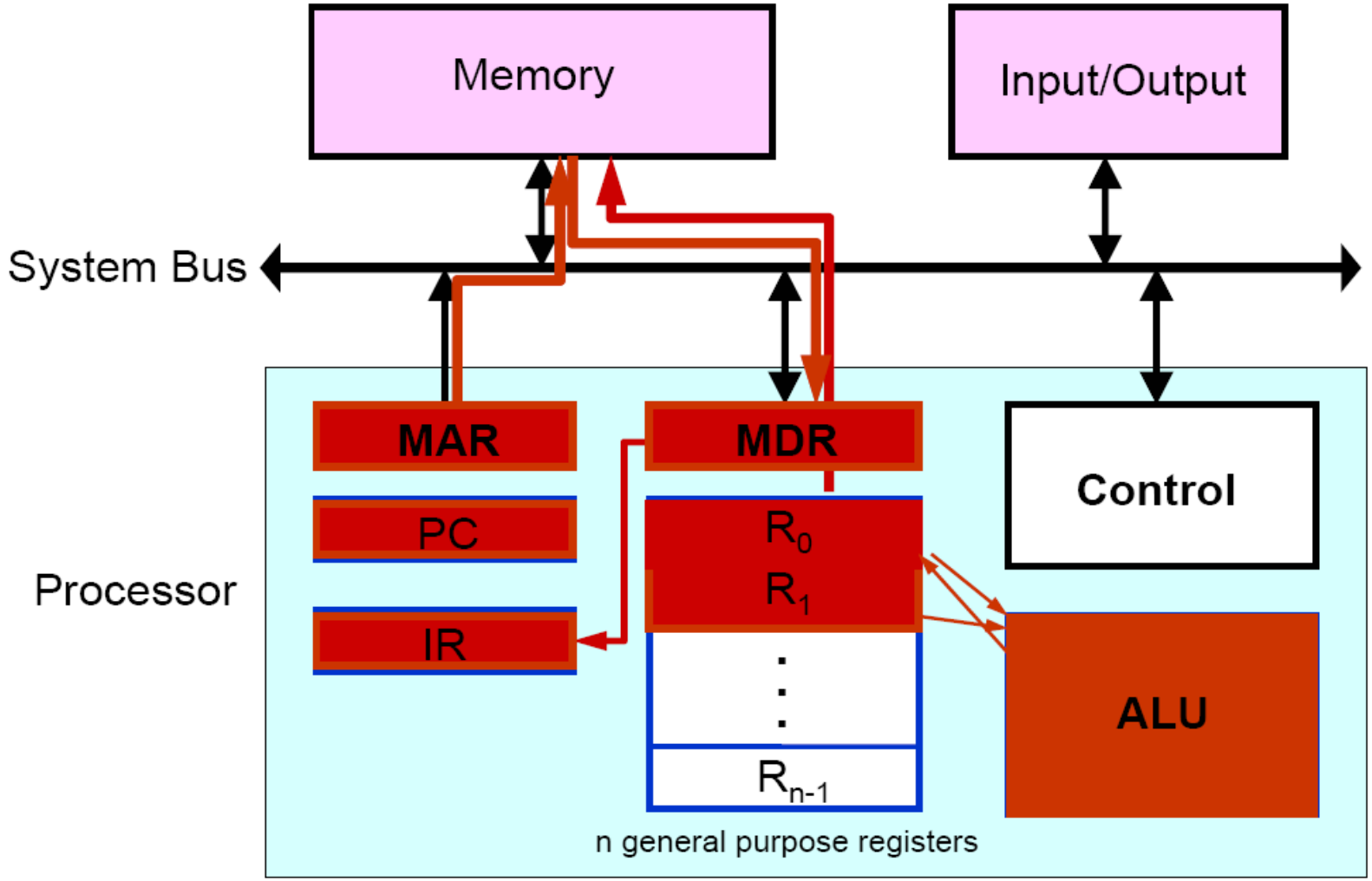


Why is the access time of the cache memory lesser than the access time of the main memory?

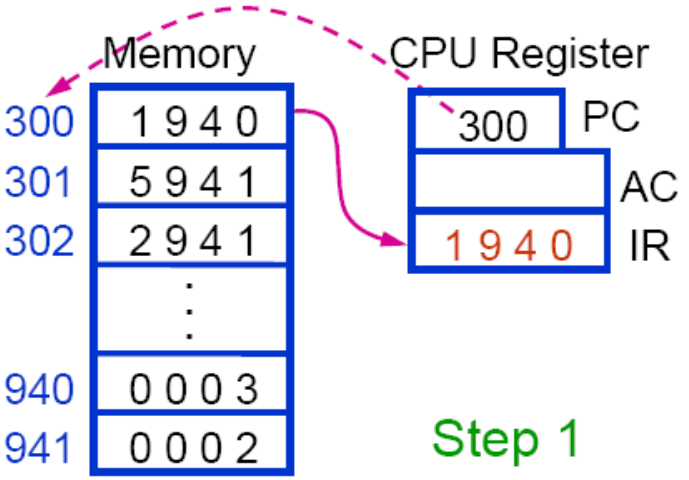
Computer Components: Top-Level View

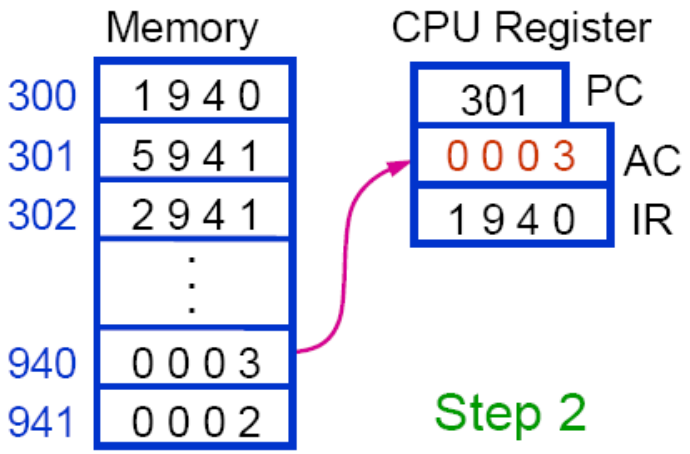


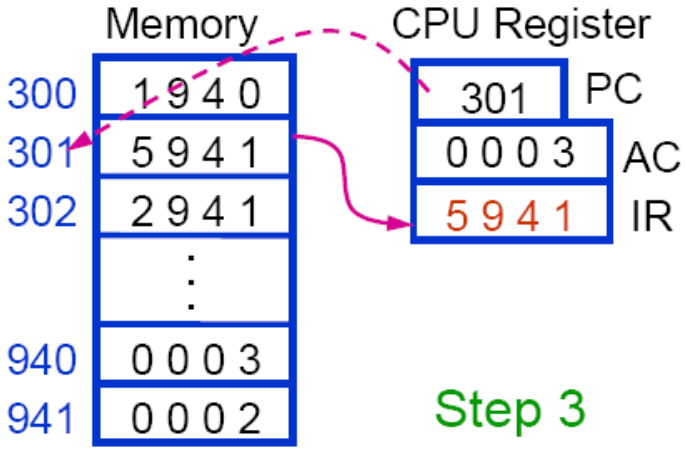
Basic Operational Concepts

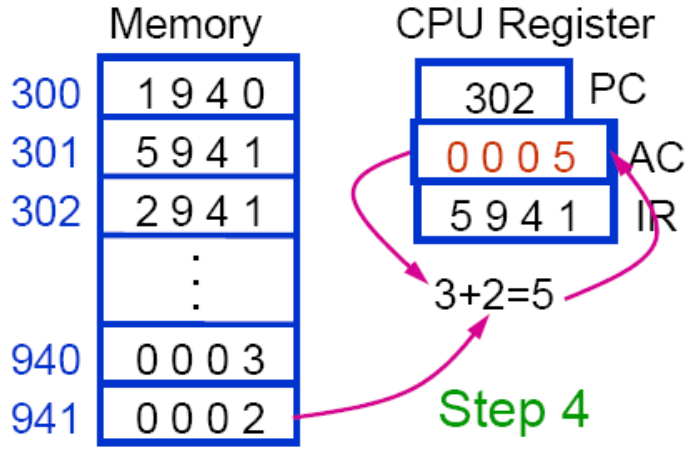


A Partial Program Execution Example

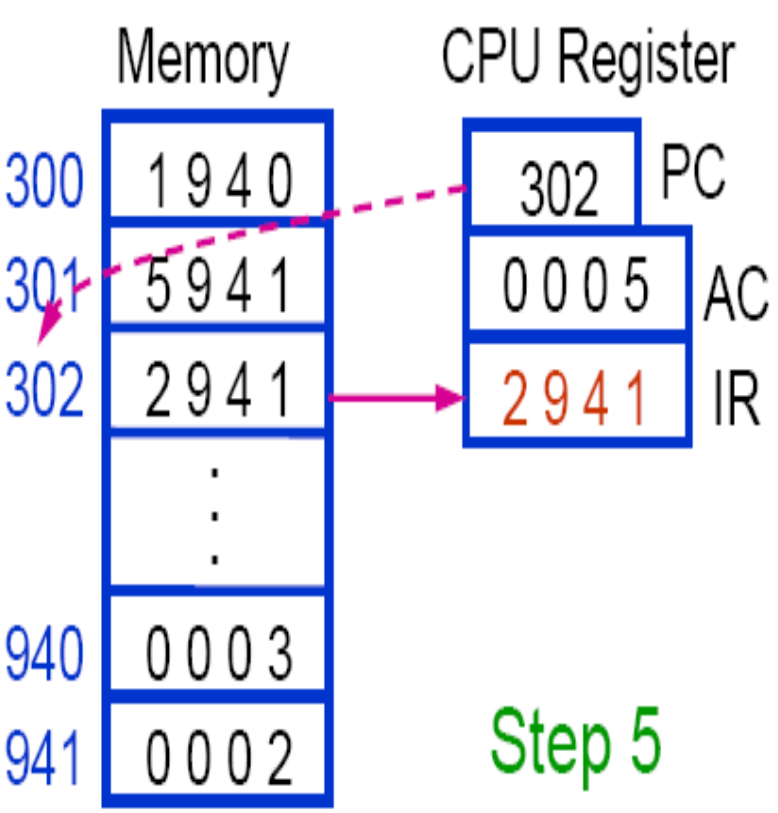


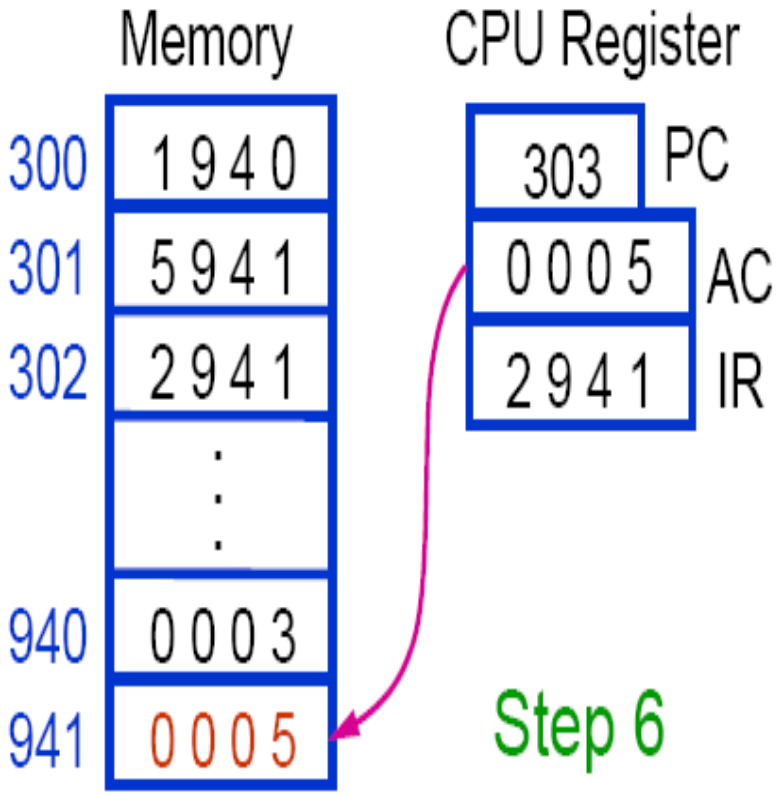






A Partial Program Execution Example





Interrupt

- Normal execution of programs may be interrupted if some device requires urgent servicing
  - To deal with the situation immediately, the normal execution of the current program must be interrupted

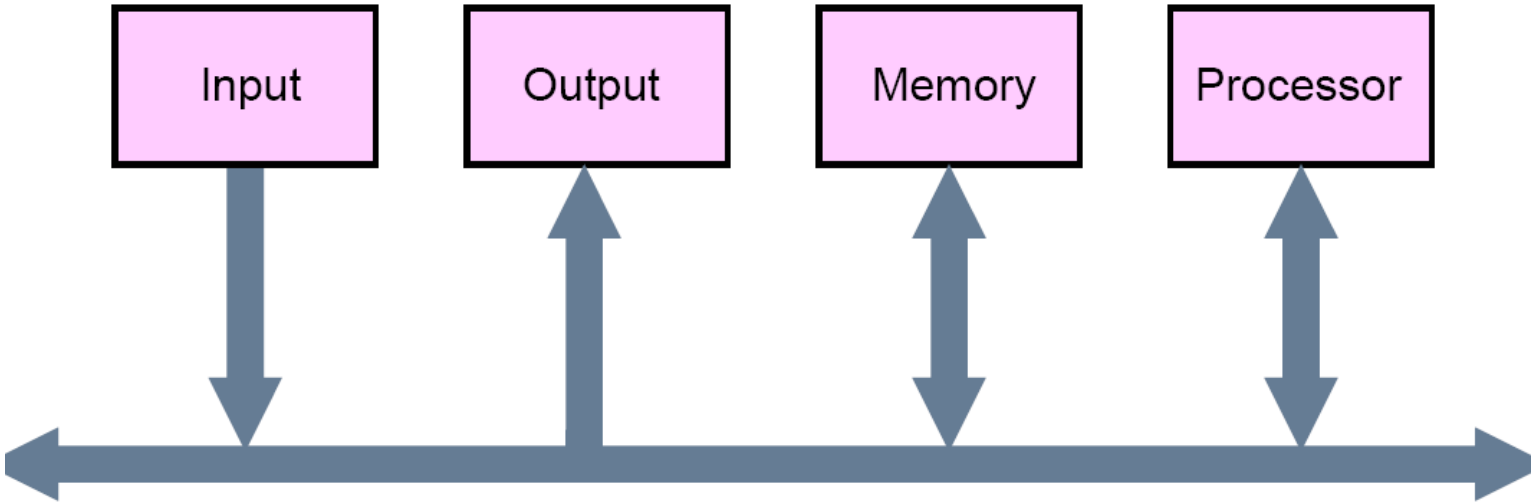
- Procedure of interrupt operation
  - The device raises an interrupt signal
  - The processor provides the requested service by executing an appropriate interrupt-service routine
  - The state of the processor is first saved before servicing the interrupt
    - Normally, the contents of the PC, the general registers, and some control information are stored in memory
  - When the interrupt-service routine is completed, the state of the processor is restored so that the interrupted program may continue

Classes of Interrupts

- Program
  - Generated by some condition that occurs as a result of an instruction execution such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space
- Timer
  - Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis
- I/O
  - Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions
- Hardware failure
  - Generated by a failure such as power failure or memory error

Bus Structures

- A group of lines that serves a connecting path for several devices is called a bus
  - In addition to the lines that carry the data, the bus must have lines for address and control purposes
  - The simplest way to interconnect functional units is to use a single bus, as shown below



Drawbacks of the Single Bus Structure

- The devices connected to a bus vary widely in their speed of operation
  - Some devices are relatively slow, such as printer and keyboard
  - Some devices are considerably fast, such as optical disks
  - Memory and processor units operate are the fastest parts of a computer
- Efficient transfer mechanism thus is needed to cope with this problem
  - A common approach is to include buffer registers with the devices to hold the information during transfers
  - An another approach is to use two-bus structure and an additional transfer mechanism
    - A high-performance bus, a low-performance, and a bridge for transferring the data between the two buses. ARMA Bus belongs to this structure

Software

- In order for a user to enter and run an application program, the computer must already contain some system software in its memory

- System software is a collection of programs that are executed as needed to perform functions such as
  - Receiving and interpreting user commands
  - Running standard application programs such as word processors, etc, or games
  - Managing the storage and retrieval of files in secondary storage devices
  - Controlling I/O units to receive input information and produce output results

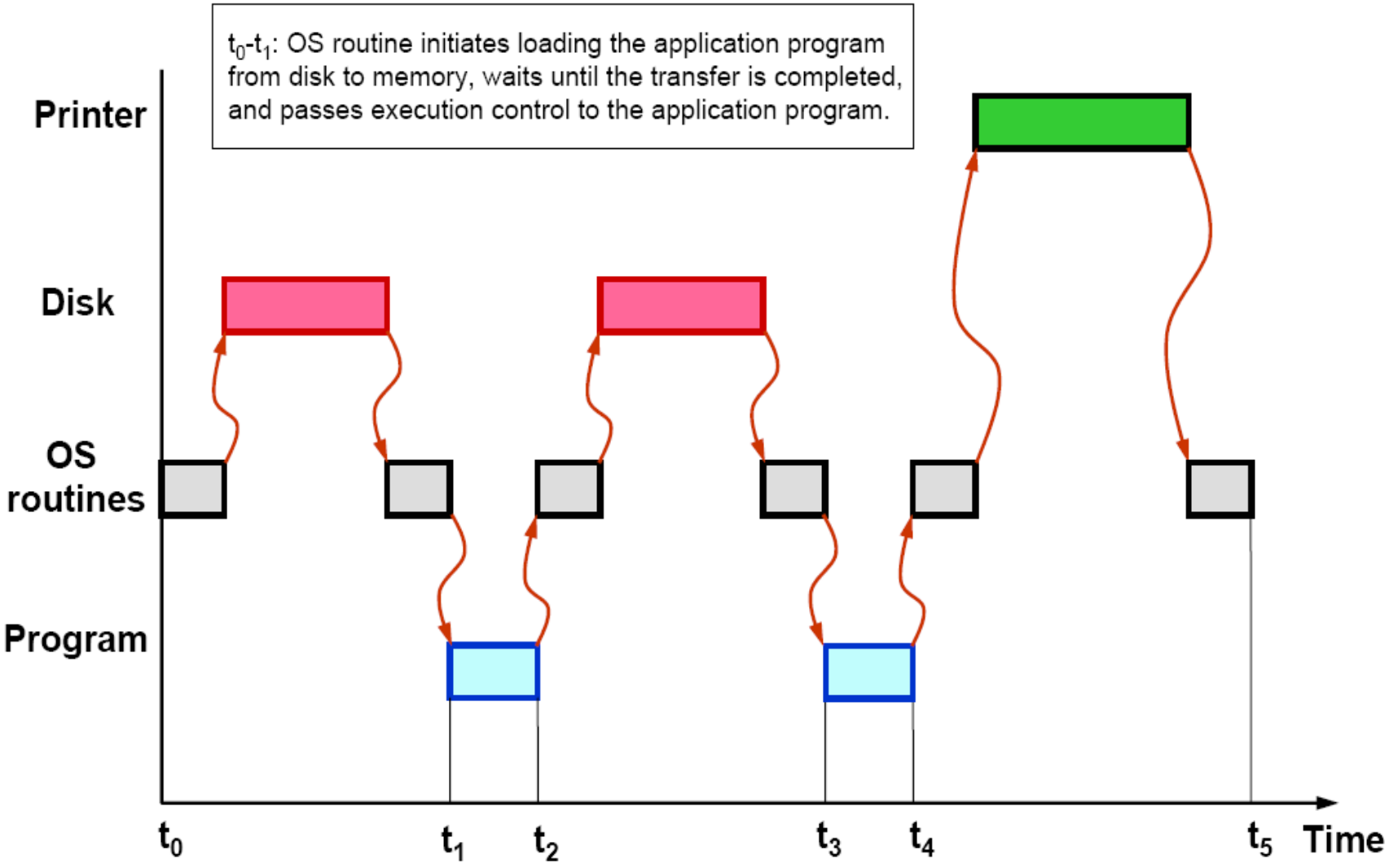
Software

- Translating programs from source form prepared by the user into object form consisting of machine instructions
- Linking and running user-written application programs with existing standard library routines, such as numerical computation packages
- System software is thus responsible for the coordination of all activities in a computing system

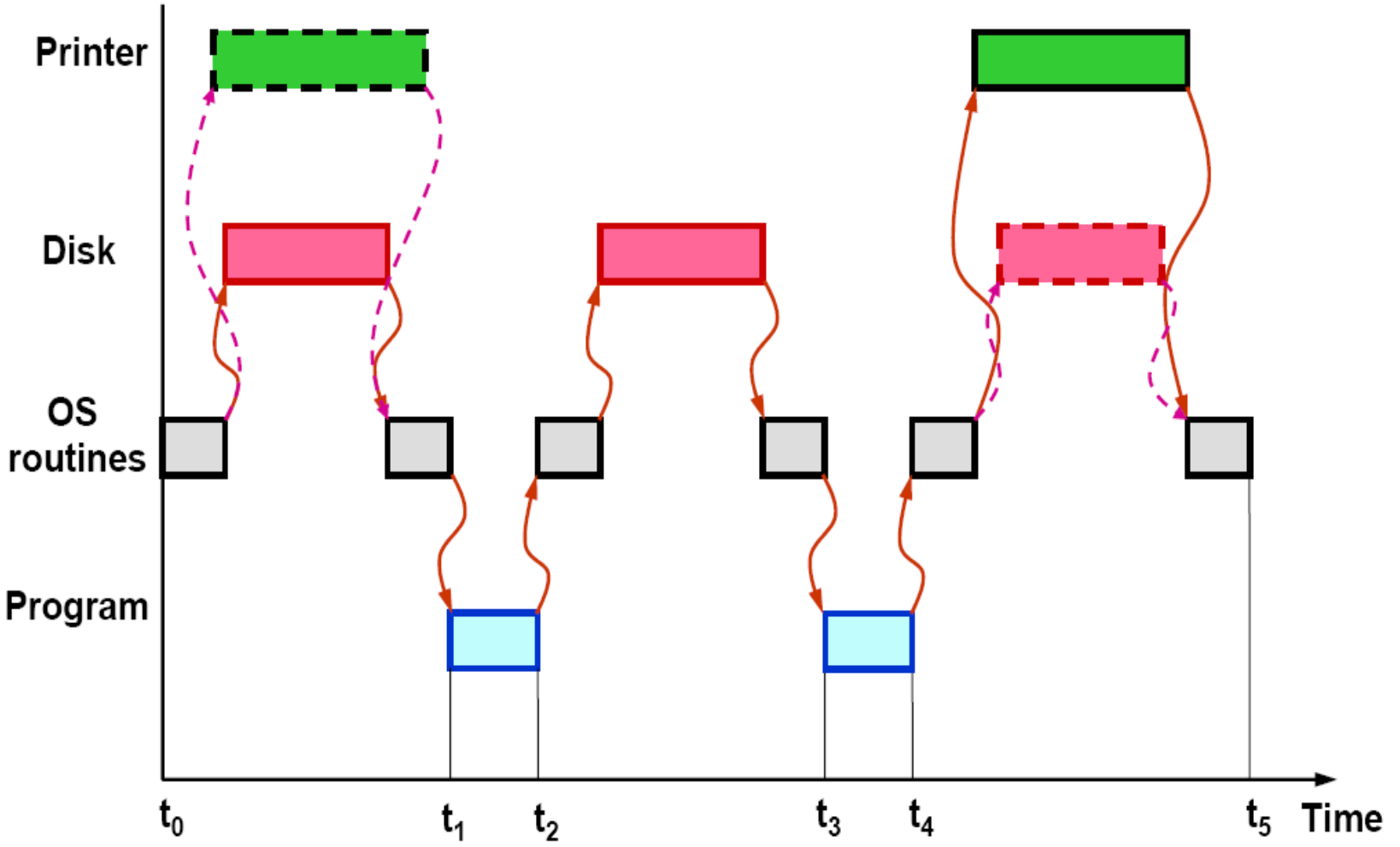
Operating System

- Operating system (OS)
  - This is a large program, or actually a collection of routines, that is used to control the sharing of and interaction among various computer units as they perform application programs
- The OS routines perform the tasks required to assign computer resource to individual application programs
  - These tasks include assigning memory and magnetic disk space to program and data files, moving data between memory and disk units, and handling I/O operations
- In the following, a system with one processor, one disk, and one printer is given to explain the basics of OS
  - Assume that part of the program's task involves reading a data file from the disk into the memory, performing some computation on the data, and printing the results

User Program and OS Routine Sharing



Multiprogramming or Multitasking



Performance

- The speed with which a computer executes programs is affected by the design of its hardware and its machine language instructions

- Because programs are usually written in a high-level language, performance is also affected by the compiler that translates programs into machine languages

- For best performance, the following factors must be considered
  - Compiler
  - Instruction set
  - Hardware design

Performance

- Processor circuits are controlled by a timing signal called a clock
  - The clock defines regular time intervals, called clock cycles
- To execute a machine instruction, the processor divides the action to be performed into a sequence of basic steps, such that each step can be completed in one clock cycle
- Let the length P of one clock cycle, its inverse is the clock rate, R=1/P
- Basic performance equation
  - T=(NxS)/R, where T is the processor time required to execute a program, N is the number of instruction executions, and S is the average number of basic steps needed to execute one machine instruction. N, S, and R are not independent parameters.

Performance

The performance parameter T for an application program is much more important to the user than the individual values of the parameters N, S, or R. To achieve high performance, the computer designer must seek ways to reduce the value of T, which means reducing N and S, and increasing R. The value of N is reduced if the source program is compiled into fewer machine instructions. The value of S is reduced if instructions have a smaller number of basic steps to perform or if the execution of instructions is overlapped. Using a higher-frequency clock increases the value or R, which means that the time required to complete a basic execution step is reduced.

Performance Improvement

- Pipelining and superscalar operation
  - Pipelining: by overlapping the execution of successive instructions
  - Superscalar: different instructions are concurrently executed with multiple instruction pipelines. This means that multiple functional units are needed
- Clock rate improvement
- Improving the integrated-circuit technology makes logic circuits faster, which reduces the time needed to complete a basic step

Performance Improvement

- Reducing amount of processing done in one basic step also makes it possible to reduce the clock period, P.
- However, if the actions that have to be performed by an instruction remain the same, the number of basic steps needed may increase
- Reduce the number of basic steps to execute
  - Reduced instruction set computers (RISC) and complex instruction set computers (CISC)

Performance Measurement

The previous discussion suggests that the only parameter that properly describes the performance of a computer is the execution time, T, for the programs of interest. Despite the conceptual simplicity of Equation 1.1, computing the value of T is not simple. Moreover, parameters such as the clock speed and various architectural features are not reliable indicators of the expected performance.

For these reasons, the computer community adopted the idea of measuring computer performance using benchmark programs. To make comparisons possible, standardized programs must be used. The performance measure is the time it takes a computer

Performance Measurement

to execute a given benchmark. Initially, some attempts were made to create artificial programs that could be used as standard benchmarks. But, synthetic programs do not properly predict performance obtained when real application programs are run.

The accepted practice today is to use an agreed-upon selection of real application programs to evaluate performance. A nonprofit organization called System Performance Evaluation Corporation (SPEC) selects and publishes representative application programs for different application domains, together with test results for many commercially available computers.

The programs selected range from game playing, compiler, and database applications to numerically intensive programs in astrophysics and quantum chemistry. In each case, the program is compiled for the computer under test, and the running time on a real computer is measured.

Performance Measurement

The same program is also compiled and run on one computer selected as a reference.

$$\text{SPEC rating} = \frac{\text{Running time on the reference computer}}{\text{Running time on the computer under test}}$$

Thus a SPEC rating of 50 means that the computer under test is 50 times as fast as the UltraSPARC10 for this particular benchmark.

The test is repeated for all the programs in the SPEC suite, and the geometric mean of the results is computed. Let SPEC_i be the rating for program i in the suite. The overall SPEC rating for the computer is given by

$$\text{SPEC rating} = \left(\prod_{i=1}^{n} \text{SPEC}_i \right)^{\frac{1}{n}}$$

where n is the number of programs in the suite.